

Exploiting Hidden Structure in Selecting Dimensions that Distinguish Vectors*

Vincent Froese^{†1}, René van Bevern^{‡2}, Rolf Niedermeier¹, and Manuel Sorge^{†1}

¹Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany,
 {vincent.froese, rolf.niedermeier, manuel.sorge}@tu-berlin.de

²Novosibirsk State University, Novosibirsk, Russia, rvb@nsu.ru

The NP-hard DISTINCT VECTORS problem asks to delete as many columns as possible from a matrix such that all rows in the resulting matrix are still pairwise distinct. Our main result is that, for binary matrices, there is a complexity dichotomy for DISTINCT VECTORS based on the maximum (H) and the minimum (h) pairwise Hamming distance between matrix rows: DISTINCT VECTORS can be solved in polynomial time if $H \leq 2\lceil h/2 \rceil + 1$, and is NP-complete otherwise. Moreover, we explore connections of DISTINCT VECTORS to hitting sets, thereby providing several fixed-parameter tractability and intractability results also for general matrices.

1 Introduction

Feature selection in a high-dimensional feature space means to choose a subset of features (that is, dimensions) such that some desirable data properties are preserved or achieved in the induced subspace. *Combinatorial* feature selection [24, 7] is a well-motivated alternative to the more frequently studied affine feature selection. While *affine* feature selection combines features to reduce dimensionality, combinatorial feature selection simply discards some features. The advantage of the latter is that the resulting reduced

*A preliminary version appeared under the title “A Parameterized Complexity Analysis of Combinatorial Feature Selection Problems” in the proceedings of the 38th International Symposium on Mathematical Foundations of Computer Science (MFCS ’13), volume 8087 of Lecture Notes in Computer Science, pages 445–456, Springer, 2013 [19]. Parts of this work originate from the first author’s master’s thesis on combinatorial feature selection [18]. This article now exclusively focuses on the DISTINCT VECTORS problem and provides all proofs in full detail. It additionally contains a new main result for DISTINCT VECTORS regarding a computational complexity dichotomy for the parameters minimum and maximum pairwise Hamming distance of the data points.

[†]Supported by Deutsche Forschungsgemeinschaft, project DAMM (NI 369/13).

[‡]Supported by Deutsche Forschungsgemeinschaft, project DAPA (NI 369/12).

feature space is easier to interpret. See Charikar et al. [7] for a more extensive discussion in favor of combinatorial feature selection. Unfortunately, combinatorial feature selection problems are typically computationally very hard to solve (NP-hard and also hard to approximate [7]), resulting in the use of heuristic approaches in practice [4, 11, 17, 22].

In this work, we adopt the fresh perspective of parameterized complexity analysis. We thus refine the known picture of the computational complexity landscape of a prominent and formally simple combinatorial feature selection problem called DISTINCT VECTORS.

DISTINCT VECTORS

Input: A matrix $S \in \Sigma^{n \times d}$ over a finite alphabet Σ with n distinct rows and $k \in \mathbb{N}$.

Question: Is there a subset $K \subseteq [d]$ of column indices with $|K| \leq k$ such that all n rows in $S|_K$ are still distinct?

Here, $S|_K$ is the submatrix containing only the columns with indices in K . In the above formulation, the input data is considered to be a matrix where the row vectors correspond to the data points and the columns represent features (dimensions). Thus, DISTINCT VECTORS constitutes the basic task to compress the data by discarding redundant or negligible dimensions without losing the essential information to tell apart all data points.

Intuitively speaking, the guiding principle of this work is to identify problem-specific parameters (quantities such as the number of dimensions to discard or the number of dimensions to keep) and to analyze how these quantities influence the computational complexity of DISTINCT VECTORS. The point here is that in relevant applications these parameters can be small, which may allow for more efficient solvability. Hence, the central question is whether DISTINCT VECTORS is computationally tractable in the case of small parameter values.

We are particularly interested in the complexity of DISTINCT VECTORS if the range of differences between data points is small. This special case occurs if the input data is in some sense homogeneous. We measure the range of differences as the gap $H - h$ between the maximum H and the minimum h of pairwise Hamming distances of rows in the input matrix.¹ We initiate the study of this measure by completely classifying the classical complexity of DISTINCT VECTORS with respect to constant values of $H - h$ on binary input matrices. For general matrices, we derive various tractability and intractability results with respect to the parameters alphabet size $|\Sigma|$, number of retained columns and number of discarded columns.

Related Work DISTINCT VECTORS is also known as the MINIMAL REDUCT problem in rough set theory [28] and it was already early proven to be NP-hard by Skowron and Rauszer [29]. Later, Charikar et al. [7] investigated the computational complexity of several problems arising in the context of combinatorial feature selection, including DISTINCT VECTORS. Seemingly unaware of Skowron and Rauszer’s work, they showed that there exists a constant c such that it is NP-hard to approximate DISTINCT VECTORS in polynomial time within a factor of $c \log d$.

¹See Section 3 for a formal definition.

Table 1: Overview of our results.

Result*	Reference
NP-hard for $ \Sigma = 2$ and $H \geq 2\lceil h/2 \rceil + 2$	Theorem 4
poly-time for $ \Sigma = 2$ and $H \leq 2\lceil h/2 \rceil + 1$	Theorem 9
W[1]-hard wrt. t for $ \Sigma = 2$ and $H \geq 4$	Corollary 2
W[2]-hard wrt. k ($ \Sigma $ unbounded)	Theorem 10
FPT wrt. (Σ , k) (no poly kernel wrt. (n, Σ , k))	Theorem 12
FPT wrt. (H, k) (for arbitrary Σ)	Theorem 13

* $|\Sigma|$: alphabet size, h (H): minimum (maximum) pairwise row Hamming distance of the input matrix, t : number of discarded columns, k : number of retained columns

Another combinatorial feature selection problem called MINIMUM FEATURE SET is a variant of DISTINCT VECTORS where not all pairs of rows have to be distinguished but only all pairs of rows from two specified subsets. This problem is known to be NP-complete for binary input data [12]. In addition, Cotta and Moscato [9] investigated the parameterized complexity of MINIMUM FEATURE SET and proved W[2]-completeness with respect to the number of selected columns even for binary matrices.

Results and Outline [Table 1](#) summarizes our results. We first focus on the case of input matrices over binary alphabets, that is $|\Sigma| = 2$, in [Section 3](#). As our main result, we completely classify the classical computational complexity of (binary) DISTINCT VECTORS according to the gap between H and h . This yields the following dichotomy: If $H \leq 2\lceil h/2 \rceil + 1$, then DISTINCT VECTORS is polynomial-time solvable, whereas it is NP-complete in all other cases. The corresponding NP-completeness proof also implies W[1]-hardness with respect to the parameter “number $t = d - k$ of columns to discard”.

In [Section 4](#) we consider general alphabets, that is, $|\Sigma| \geq 2$. We prove that, here, DISTINCT VECTORS is W[2]-hard with respect to the number k of retained columns if the alphabet size is unbounded. Moreover, DISTINCT VECTORS cannot be solved in $d^{o(k)}(nd)^{O(1)}$ time, unless W[1] = FPT (which is strongly believed not to be the case [15]). In contrast to these hardness results, we develop polynomial-time data reduction algorithms and show fixed-parameter tractability by providing superexponential-size problem kernelizations with respect to the combined parameters $(|\Sigma|, k)$ and (H, k) . We also exclude polynomial-size problem kernels with respect to the parameter combination $(n, |\Sigma|, k)$ based on the hypothesis that $\text{NP} \not\subseteq \text{coNP/poly}$ (which is believed to be true, since otherwise the polynomial hierarchy collapses to its third level). Finally, as a simple observation, we also give a linear-time factor- H approximation algorithm.

Our notation is explained in [Section 2](#). [Section 5](#) concludes with some challenges for future research.

2 Preliminaries

Notation For $n \in \mathbb{N}$, let $[n] := \{1, \dots, n\}$. The set of all size- k subsets of a set X is denoted by $\binom{X}{k}$. In the following, we consider finite alphabets $\Sigma \subseteq \mathbb{Q}_0^+$. We denote by $S = (s_{ij}) \in \Sigma^{n \times d}$ the matrix with n rows and d columns, where $s_{ij} \in \Sigma$ denotes the entry in the i -th row and the j -th column. We denote the i -th row vector by s_i and the j -th column vector by s_{*j} . For subsets $I \subseteq [n]$ and $J \subseteq [d]$ of row and column indices, we write $S[I, J] := (s_{ij})_{(i,j) \in I \times J}$ for the $|I| \times |J|$ submatrix of S containing only the rows with indices in I and the columns with indices in J . We use the abbreviation $S_{\setminus J} := S[[n], J]$ for the submatrix containing all rows but only the columns in J and we say that the columns in $[d] \setminus J$ are *discarded* (or deleted). For a vector $x \in \Sigma^d$, we denote by $(x)_j \in \Sigma$ the j -th entry of x . The null vector is denoted by $\mathbf{0} := (0, \dots, 0)$.

Throughout this work, we assume that arithmetic operations such as additions and comparisons of numbers can be done in $O(1)$ time (that is, we use the RAM model [27]).

Parameterized Complexity We assume the reader to be familiar with the basic concepts from classical complexity theory, such as NP-hardness and polynomial-time reductions [20, 27]. The computational complexity of a parameterized problem is measured in terms of two quantities: one is the input size, the other is the *parameter* (usually a positive integer). A parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is called *fixed-parameter tractable* with respect to a parameter k if it can be solved in $f(k) \cdot |I|^{O(1)}$ time, where f is a computable function only depending on k , and $|I|$ is the size of the input instance I . A *problem kernel* for a parameterized problem P is a polynomial-time self-reduction, that is, given an instance (I, k) , it outputs another instance (I', k') of P such that $|I'| + k' \leq g(k)$ for some computable function g depending only on k , and (I, k) is a yes-instance of P if and only if (I', k') is a yes-instance of P . The function g is called the size of the problem kernel. If g is a polynomial, then we speak of a polynomial kernel. Existence of a problem kernel is equivalent to fixed-parameter tractability [10, 15, 16, 26].

A *parameterized reduction* from a parameterized problem P to another parameterized problem P' is a function that, given an instance (I, k) of P , computes in $f(k) \cdot |I|^{O(1)}$ time an instance (I', k') (with k' only depending on k) such that (I, k) is a yes-instance of P if and only if (I', k') is a yes-instance of P' . The two basic complexity classes for showing (presumable) fixed-parameter intractability are called W[1] and W[2]; there is good complexity-theoretic reason to believe that W[1]-hard and W[2]-hard problems are not fixed-parameter tractable [10, 15, 16, 26].

3 Binary Matrices and the Range of Differences

Throughout this section, we focus on instances with a binary input alphabet, say, without loss of generality, $\Sigma = \{0, 1\}$. We analyze the computational complexity with respect to the range of differences between input data points. To this end, we consider instances where the Hamming distance of each pair of rows lies within a prespecified range. In other words, the number of columns in which a given pair of rows differs shall be bounded

from below and above by some constants $\alpha, \beta \in \mathbb{N}$. We first give the formal definitions and then completely classify the classical complexity of DISTINCT VECTORS with respect to the gap between α and β . The NP-complete cases are given in [Section 3.1](#) and the polynomial cases in [Section 3.2](#). The formal definitions for our setup are the following.

Definition 1 (Weight). *For a vector $x \in \{0, 1\}^d$, we denote by $W_x := \{j \in [d] \mid (x)_j = 1\}$ the set of indices where x equals 1 and we call $w(x) := |W_x|$ the weight of x .*

Definition 2 (Hamming Distance). *For vectors $x, y \in \Sigma^d$, let $D_{xy} := \{j \in [d] \mid (x)_j \neq (y)_j\}$ be the set of indices where x and y differ and let $\Delta(x, y) := |D_{xy}|$ denote the Hamming distance of x and y .*

Note that, for $x, y \in \{0, 1\}^d$, it holds $D_{xy} = (W_x \cup W_y) \setminus (W_x \cap W_y)$ and thus $\Delta(x, y) = w(x) + w(y) - 2|W_x \cap W_y|$. For a DISTINCT VECTORS instance $(S \in \Sigma^{n \times d}, k)$, we define the parameters *minimum pairwise row Hamming distance* $h := \min_{i \neq j \in [n]} \Delta(s_i, s_j)$ and *maximum pairwise row Hamming distance* $H := \max_{i \neq j \in [n]} \Delta(s_i, s_j)$. To conveniently state our results, let us now define a variant of DISTINCT VECTORS with minimum pairwise row Hamming distance α and maximum pairwise row Hamming distance β :

BINARY (α, β) -DISTINCT VECTORS

Input: A matrix $S \in \{0, 1\}^{n \times d}$ with n distinct rows such that $\alpha = h \leq H = \beta$, and $k \in \mathbb{N}$.

Question: Is there a subset $K \subseteq [d]$ of column indices with $|K| \leq k$ such that all rows in $S|_K$ are still distinct?

Intuitively, if the matrix consists of rows that are all “similar” to each other, one could hope to be able to solve the instance efficiently since there are at most β columns to choose from in order to distinguish two rows. The minimum pairwise row Hamming distance α plays a dual role in the sense that, if α is large, then each pair of rows differs in many columns, which also could make the instance easily solvable. The following theorems, however, show that this intuition is somewhat deceptive in the sense that BINARY (α, β) -DISTINCT VECTORS is NP-hard even for small constants α and β . Despite this intimidating NP-hardness result, we perform a closer inspection of the relation between the minimum and maximum pairwise row Hamming distance and show that it is possible to solve some cases in polynomial time for arbitrarily large constants α, β . These results are obtained by applying combinatorial arguments from extremal set theory revealing a certain structure of the input matrix if the values of α and β are close to each other, that is, the range of differences is small. Analyzing this structure, we can show how to find solutions in polynomial time.

[Figure 1](#) depicts the (non-parameterized) computational complexity landscape for BINARY (α, β) -DISTINCT VECTORS with respect to α and β , indicating the border of hardness. In the following, we will step by step develop the results exhibited in [Figure 1](#).

3.1 NP-Completeness for Heterogeneous Data

As a starting point, we prove the NP-completeness of the case $\alpha = 2, \beta = 4$.

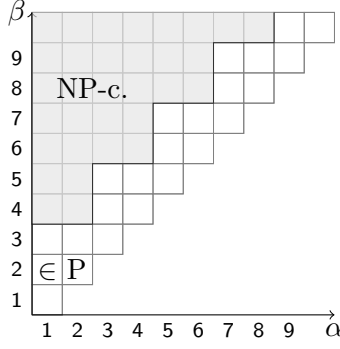


Figure 1: Overview of the complexity of BINARY (α, β) -DISTINCT VECTORS. Gray cells correspond to NP-complete cases, whereas white cells are polynomial-time solvable cases.

Theorem 1. BINARY $(2, 4)$ -DISTINCT VECTORS is NP-complete.

Proof. It is easy to check that DISTINCT VECTORS is in NP. To prove NP-hardness, we give a polynomial-time many-one reduction from a special variant of the INDEPENDENT SET problem in graphs, which is defined as follows.

DISTANCE-3 INDEPENDENT SET

Input: An undirected graph $G = (V, E)$ and $k \in \mathbb{N}$.

Question: Is there a subset of vertices $I \subseteq V$ of size at least k such that each pair of vertices from I has distance at least three?

Here, the distance of two vertices is the number of edges contained in a shortest path between them. DISTANCE-3 INDEPENDENT SET is known to be NP-complete by a reduction from the NP-complete INDUCED MATCHING problem [5].

Our reduction works as follows: Let $(G = (V, E), k)$ with $|V| = n$ and $|E| = m$ be an instance of DISTANCE-3 INDEPENDENT SET and let $Z \in \{0, 1\}^{m \times n}$ be the incidence matrix of G with rows corresponding to edges and columns to vertices, that is, $z_{ij} = 1$ means that the i -th edge contains the j -th vertex. We assume that G contains no isolated vertices since they are always contained in a maximum distance-3 independent set and can thus be removed. We further assume that G contains at least four edges of which at least two are disjoint. Otherwise, G is either of constant size or a star, for which a maximum distance-3 independent set consists of only a single vertex. Hence, we can solve these cases in polynomial time and return a trivial yes- or no-instance.

The matrix $S \in \{0, 1\}^{(m+1) \times n}$ of the BINARY $(2, 4)$ -DISTINCT VECTORS instance (S, k') is defined as follows: $s_i := z_i$ for all $i \in [m]$ and $s_{m+1} := \mathbf{0}$. The desired solution size is set to $k' := n - k$. Notice that each row in Z contains exactly two 1's and no two rows are equal since G contains no multiple edges. Moreover, by assumption, there exists a pair of rows with Hamming distance four since G contains a pair of disjoint edges. Since S contains the null vector as a row, it follows that $h = 2$ and $H = 4$. The instance (S, k') can be computed in $O(nm)$ time.

The correctness of the reduction is due to the following argument. The instance (G, k)

is a yes-instance if and only if there is a set $I \subseteq V$ of size exactly k such that every edge in G has at least one endpoint in $V \setminus I$ and no vertex in $V \setminus I$ has two neighbors in I . In other words, the latter condition says that no two edges with an endpoint in I share the same endpoint in $V \setminus I$. Equivalently, for the subset K of columns corresponding to the vertices in $V \setminus I$, it holds that all rows in $S[[m], K]$ contain at least one 1 and no two rows contain only a single 1 in the same column. This is true if and only if K is a solution for (S, k') because s_{m+1} equals the null vector and thus two rows in $S|_K$ can only be identical if either they consist of 0's only or contain only a single 1 in the same column. Furthermore, $|K| = |V \setminus I| = n - k = k'$. \square

We remark that from a W[1]-hardness result for INDUCED MATCHING parameterized by the number of vertices in the induced subgraph [25], we can infer W[1]-hardness for DISTANCE-3 INDEPENDENT SET with respect to the solution size k . Since the proof of Theorem 1 actually provides a parameterized reduction from DISTANCE-3 INDEPENDENT SET parameterized by k to DISTINCT VECTORS parameterized by the number of columns to discard (which is $d - k' = n - (n - k) = k$ in the above reduction), we have the following:

Corollary 2. BINARY $(2, 4)$ -DISTINCT VECTORS is W[1]-hard with respect to the number $t := d - k$ of discarded columns.

Note, however, that the reduction in the proof of Theorem 1 is not a parameterized reduction with respect to the number $k' = n - k$ of retained columns since k' does not solely depend on k but also on the number n of vertices. Hence, we cannot infer W[1]-hardness with respect to k' . In fact, we will show in Section 4 that DISTINCT VECTORS allows a problem kernel with respect to the number of retained columns for binary alphabets.

We will now give polynomial-time reductions from BINARY $(2, 4)$ -DISTINCT VECTORS to certain other cases of BINARY (α, β) -DISTINCT VECTORS with different bounds on the minimum and maximum Hamming distance. Using Theorem 1 as an anchor point, we can then derive all remaining NP-completeness results in Figure 1. The reductions will mainly build on some padding arguments, that is, starting from a given input matrix, we expand it by adding new columns and rows such that we achieve the desired constraints on the Hamming distances without changing the actual answer to the original instance. To start with, we define a type of column vectors which can be used for padding an input matrix without changing the answer to the original instance, that is, such “padding columns” are not contained in an optimal solution. Informally, a column j is *inessential* if all rows could still be distinguished by the same number of columns without selecting j . The formal definition is the following.

Definition 3. For a matrix $S \in \Sigma^{n \times d}$, a column $j \in [d]$ is called *inessential* if the following two conditions are fulfilled:

- (1) There exists a row $i \in [n]$ such that column j exactly distinguishes row i from all other rows, that is, $s_{ij} \neq s_{lj}$ and $s_{lj} = s_{l'j}$ holds for all $l, l' \in [n] \setminus \{i\}$.
- (2) All rows in $S_{[d] \setminus \{j\}}$ are still distinct.

Note that for binary matrices, Condition (1) of [Definition 3](#) can only be fulfilled by column vectors that contain either a single 1 or a single 0, that is, the column vectors of weight 1 or $n - 1$.

Next, we show that, for any inessential column in a given input matrix, we can assume that this column is not contained in a solution for the DISTINCT VECTORS instance.

Lemma 3. *Let $(S \in \{0, 1\}^{n \times d}, k)$ be a DISTINCT VECTORS instance with an inessential column $j \in [d]$. It holds that (S, k) is a yes-instance if and only if $(S_{[d] \setminus \{j\}}, k)$ is a yes-instance.*

Proof. It is clear that the “if” part of the statement holds; let us consider the “only if” part. To this end, assume that there is a solution set $K \subseteq [d]$ of columns for (S, k) with $j \in K$. Since column j exactly distinguishes row i from all other rows and no other pair of rows, it follows that $K' := K \setminus \{j\}$ is a solution for $(S[[n] \setminus \{i\}, [d] \setminus \{j\}], k - 1)$. But then, there also exists a solution $K'' \subseteq [d] \setminus \{j\}$ for $(S[[n], [d] \setminus \{j\}], k)$. This is true because row i equals at most one other row l in $S[[n], K']$ since all rows in $S[[n] \setminus \{i\}, K']$ are distinct. Row i can thus be distinguished from row l by a column $j' \in [d] \setminus \{j\}$ with $s_{ij'} \neq s_{lj'}$, which exists because column j is inessential, and thus, by definition, all rows in $S[[n], [d] \setminus \{j\}]$ are distinct. Hence, $K'' := K' \cup \{j'\}$ is a solution for (S, k) . \square

Note that, due to [Lemma 3](#), adding inessential columns to a given input matrix yields an equivalent DISTINCT VECTORS instance. Hence, for the binary case, any construction that only adds column vectors which either contain a single 1 or a single 0 to the input matrix yields an equivalent instance since these columns are clearly inessential. Following this basic idea, the proof of the following theorem shows which Hamming distances can be generated from a given input matrix by adding inessential columns.

Theorem 4. BINARY (α, β) -DISTINCT VECTORS is NP-complete for all

- $\beta \geq \alpha + 2$ if α is even, and
- $\beta \geq \alpha + 3$ if α is odd.

Proof. In the following, we give polynomial-time many-one reductions from BINARY $(2, 4)$ -DISTINCT VECTORS. To this end, let $(S \in \{0, 1\}^{n \times d}, k)$ be the BINARY $(2, 4)$ -DISTINCT VECTORS instance as constructed in the proof of [Theorem 1](#). Recall that this matrix S contains the null row vector, say $s_n = \mathbf{0}$, and all other rows have weight two, $w(s_i) = 2$ for all $i \in [n - 1]$. Moreover, there exists a pair of rows with Hamming distance four. Assume, without loss of generality, that the first two rows s_1 and s_2 have Hamming distance $\Delta(s_1, s_2) = 4$. [Figure 2a](#) depicts an example of such a matrix. In the following, let $D_j = D_{s_{*j}} \subseteq [n]$ denote the set of row indices where column vector s_{*j} equals 1. Further, for $i \in \mathbb{N}$ and $I \subseteq [i]$, let $\mathbf{1}_I^i \in \{0, 1\}^i$ denote the size- i vector that has 1-entries at all indices in I and 0-entries elsewhere.

We prove the theorem in two steps: First, the case $\alpha = 1$, $\beta = 4 + b$ for some $b \geq 0$, and second, the case $\alpha = 2 + a$, $\beta = 4 + 2\lceil a/2 \rceil + b$ for some $a, b \geq 0$. Note that these two cases together yield the statement of the theorem.

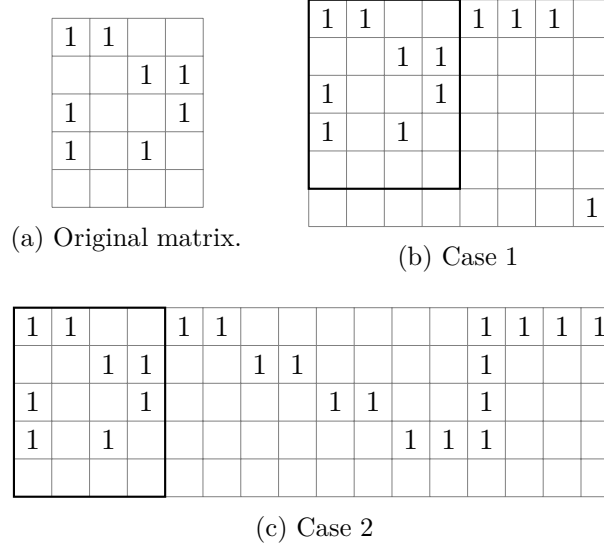


Figure 2: Example of the construction for $\alpha = 3, \beta = 3$.

Case 1 ($\alpha = 1, \beta = 4 + b, b \geq 0$). We define the instance (S', k') as follows: The column vectors of the matrix $S' \in \{0, 1\}^{(n+1) \times (d+b+1)}$ are

$$s'_{*j} := \begin{cases} \mathbf{1}_{D_j}^{n+1}, & j \in \{1, \dots, d\}, \\ \mathbf{1}_{\{1\}}^{n+1}, & j \in \{d+1, \dots, d+b\}, \\ \mathbf{1}_{\{n+1\}}^{n+1}, & j = d+b+1. \end{cases}$$

We set $k' := k + 1$. An example of the constructed instance is shown in [Figure 2b](#). It is not hard to check that the rows of S' indeed fulfill the constraints on the Hamming distances:

$$\begin{aligned} h' &:= \min_{i \neq j \in [n+1]} \Delta(s_i, s_j) = \Delta(s'_n, s'_{n+1}) = 1, \\ H' &:= \max_{i \neq j \in [n+1]} \Delta(s_i, s_j) = \Delta(s'_1, s'_2) = \Delta(s_1, s_2) + b = 4 + b. \end{aligned}$$

As regards correctness, observe first that any solution contains the column index $d+b+1$ because the row vectors s'_n and s'_{n+1} only differ in this column. Since this column also distinguishes row s_{n+1} from all other rows and no other pair of rows in S' , it follows that (S', k') is a yes-instance if and only if $(S'_{[d+b]}, k)$ is a yes-instance. Due to [Lemma 3](#), this is the case if and only if (S, k) is a yes-instance.

Case 2 ($\alpha = 2 + a, \beta = 4 + 2\lceil a/2 \rceil + b, a, b \geq 0$). We define the instance (S', k') as follows: Starting with $S' := S$, we add $\lceil a/2 \rceil$ copies of the column vector $\mathbf{1}_{\{i\}}^n$ for each $i \in [n-1]$ to S' . Moreover, we add $\lfloor a/2 \rfloor$ copies of the column vector $\mathbf{1}_{[n-1]}^n$ to S' . Finally, we add b copies of the column vector $\mathbf{1}_{\{1\}}^n$ to S' and set $k' = k$. [Figure 2c](#) shows

an example of the construction. Indeed, we have the following Hamming distances:

$$\begin{aligned}\Delta(s'_n, s'_1) &= 2 + a + b, \\ \Delta(s'_n, s'_j) &= 2 + a, \\ \Delta(s'_1, s'_j) &= \Delta(s_1, s_j) + 2\lceil a/2 \rceil + b, \\ \Delta(s'_j, s'_{j'}) &= \Delta(s_j, s_{j'}) + 2\lceil a/2 \rceil,\end{aligned}$$

for all $j, j' \in \{2, \dots, n-1\}$, $j \neq j'$. Thus, it holds $h' = 2 + a$ and $H' = 4 + 2\lceil a/2 \rceil + b$. Since all row vectors in S are distinct and since we only added columns which distinguish exactly one row from all others, the correctness follows due to [Lemma 3](#). \square

[Theorem 4](#) yields the NP-completeness of BINARY (α, β) -DISTINCT VECTORS for all $\beta \geq \alpha + 2$ (for even α) and $\beta \geq \alpha + 3$ (for odd α), that is, for a given instance with fixed minimum pairwise row Hamming distance α , it is possible to increase the maximum pairwise row Hamming distance β arbitrarily without changing the answer to the instance. On the contrary, however, it seems impossible to construct an equivalent instance where only the minimum pairwise row Hamming distance is increased. Indeed, in the following, we show polynomial-time solvability for the case $\alpha \geq 2\lfloor \beta/2 \rfloor - 1$.

3.2 Polynomial-Time Solvability for Homogeneous Data

The polynomial-time algorithm for homogeneous is based on the observation that, for small differences between the values of minimum and maximum pairwise row Hamming distance, the input matrix is either highly structured or bounded in size by a constant depending only on the maximum pairwise row Hamming distance. This structure in turn guarantees that the instance is easily solvable. Before proving the theorem, we start with some basic results.

First, we show that there is a linear-time preprocessing of a given input matrix such that the resulting matrix contains the null vector as a row and no two column vectors are identical.

Lemma 5. *For a given DISTINCT VECTORS instance $I = (S \in \{0,1\}^{n \times d}, k)$ one can compute in $O(nd)$ time an equivalent DISTINCT VECTORS instance $I' := (S' \in \{0,1\}^{n \times d'}, k)$ such that S' contains the null vector $\mathbf{0} \in \{0\}^{d'}$ as a row, the number d' of columns of S' is at most d , and no two column vectors of S' are identical (implying $d' \leq 2^n$).*

Proof. From an instance $I = (S, k)$, we compute S' as follows: First, in order to have the null vector $\mathbf{0}$ as a row, we consider an arbitrary row vector, say s_1 , and iterate over all columns j . If $s_{1j} = 1$, then we exchange all 1's and 0's in column j . Then, we sort the columns of S lexicographically in $O(nd)$ time (using radix sort). We iterate over all columns again and check for any two successive column vectors whether they are identical and, if so, remove one of them. This ensures that all remaining column vectors are different, which implies that there are at most 2^n . Thus, in $O(nd)$ time, we end up with a matrix S' containing at most 2^n columns, where $s'_1 = \mathbf{0}$. Clearly, reordering

1	2	3	4	5	6	7	
1				1			$\mathcal{W}_1 = \{\{3\}\}$
1			1			1	$\mathcal{W}_2 = \{\{1, 5\}, \{6, 7\}\}$
		1					
	1		1	1			$\mathcal{W}_3 = \{\{1, 4, 7\}, \{2, 4, 5\}\}$
					1	1	

Figure 3: An example of a binary matrix (left) containing rows of weight one, two, and three. The corresponding row systems are written on the right.

columns, removing identical columns, as well as exchanging 1's and 0's in a column does not change the answer to the original instance. \square

We henceforth assume all input instances to be already preprocessed according to [Lemma 5](#). In fact, we can extend [Lemma 5](#) by removing also inessential columns (recall [Definition 3](#)), that is, we can use the following data reduction rule.

Reduction Rule 1. *Let (S, k) be a DISTINCT VECTORS instance. If S contains an inessential column, then delete this column from S .*

[Lemma 3](#) guarantees the correctness² of [Reduction Rule 1](#). Exhaustive application of [Reduction Rule 1](#) can be done as follows: First, we determine in $O(nd)$ time which columns fulfill Condition (1) of [Definition 3](#). Recall that these are exactly the weight-1 and weight- $(n-1)$ columns of which there can be at most $\min\{n, d\}$ after the preprocessing according to [Lemma 5](#). For each of these candidate columns j , we check in $O(nd)$ time whether Condition (2) also holds, that is, whether all row vectors are still distinct without column j , by lexicographically sorting the rows of the matrix without column j . The overall running time is thus in $O(\min\{n, d\} \cdot nd)$.

We now turn towards proving polynomial-time solvability of BINARY (α, β) -DISTINCT VECTORS for $\alpha \geq 2\lfloor \beta/2 \rfloor - 1$. The proof uses some results from extremal combinatorics concerning certain set systems. We refer the reader to the book by Jukna [23, Chapter 6] for an introduction into this topic. To start with, we introduce the necessary concepts and notation. Recall [Definition 1](#), where we defined the set W_{s_i} of column indices where row i equals 1. In the following, for a given input matrix S and a given set of row indices I , we will consider the *column system* of I , that is, the system containing the sets W_{s_i} of column indices of all row vectors with indices in I .

Definition 4. *For a matrix $S \in \{0, 1\}^{n \times d}$ and a subset $I \subseteq [n]$ of row indices, let $\mathcal{W}(I) := \{W_{s_i} \mid i \in I\}$ denote the column system of I containing the sets W_{s_i} of column indices for all rows in I . For $\omega \in [d]$, let $I_\omega := \{i \in [n] \mid w(s_i) = \omega\}$ be the set of indices of the weight- ω rows and let $\mathcal{W}_\omega := \mathcal{W}(I_\omega)$ be the column system of the weight- ω rows.*

[Figure 3](#) illustrates [Definition 4](#). Note that in order to distinguish all rows of weight ω from each other, we only have to consider those columns which appear in some of the sets contained in the column system \mathcal{W}_ω since the weight- ω rows only differ in these

²A reduction rule is correct if it transforms yes-instances and only yes-instances into yes-instances.

columns. Thus, in order to find subsolutions for the weight- ω rows, the structure of \mathcal{W}_ω , especially the pairwise intersections of the contained sets, will be very important for us. Therefore, we make use of two general combinatorial concepts of set systems [23], the first of which defines a system of sets that pairwise intersect in the same number of elements, whereas the second concept describes the even stronger condition that all pairwise intersections contain the same elements.

Definition 5 (Weak Δ -system). *A family $\mathcal{F} = \{S_1, \dots, S_m\}$ of m different sets is called a weak Δ -system if there is some $\lambda \in \mathbb{N}$ such that $|S_i \cap S_j| = \lambda$ for all $i \neq j \in [m]$.*

Definition 6 (Strong Δ -system). *A strong Δ -system (or sunflower) is a weak Δ -system $\{S_1, \dots, S_m\}$ such that $S_i \cap S_j = C$ for all $i \neq j \in [m]$ and some set C called the core. The sets $\tilde{S}_i := S_i \setminus C$ are called petals.*

As a first case, the following lemma illustrates the merit of the above definitions showing that any DISTINCT VECTORS instance can easily be solved if the underlying column system of all non-zero-weight rows forms a sunflower.

Lemma 6. *Let $I := (S \in \{0, 1\}^{n \times d}, k)$ be a DISTINCT VECTORS instance such that $\mathcal{W} := \bigcup_{\omega \geq 1} \mathcal{W}_\omega$ forms a sunflower (note that $\mathcal{W}_0 = \emptyset \notin \mathcal{W}$). Then, I is a yes-instance if and only if $k \geq |\mathcal{W}|$. Moreover, any solution intersects at least all but one of the petals of \mathcal{W} .*

Proof. Recall that we assume the instance I to be already preprocessed according to Lemma 5. Hence, we can assume without loss of generality, that in S , $s_n = \mathbf{0}$ and no two column vectors are equal; assume further that $\mathcal{W} = \{W_{s_1}, \dots, W_{s_{n-1}}\}$ is a sunflower with core C . An example is depicted in Figure 4.

Recall that any solution K fulfills $K \cap D_{ij} \neq \emptyset$ for all $i \neq j \in [n]$, where D_{ij} is the set of column indices in which the row vectors s_i and s_j differ. Assume towards a contradiction that $K \subseteq [d]$ with $|K| < n - 1$ is a solution. If $K \cap C = \emptyset$, then K only intersects the petals. Since the petals are pairwise disjoint, it follows that there exists an $i \in [n - 1]$ such that $K \cap W_i = K \cap D_{in} = \emptyset$, which shows that K cannot be solution. If $K \cap C \neq \emptyset$, then K intersects at most $n - 3$ of the $n - 1$ petals in \mathcal{W} . Hence, there exist $i, j \in [n - 1]$ with $i \neq j$ such that $K \cap (\tilde{W}_i \cup \tilde{W}_j) = K \cap D_{ij} = \emptyset$. Hence, K cannot be a solution. It remains to show that there is always a solution of size $|\mathcal{W}| = n - 1$. To this end, let K contain an arbitrary element from each non-empty petal and, if there is an empty petal, also an arbitrary element from the core C . Clearly, K is a solution of size $n - 1$. \square

According to Lemma 6, identifying sunflower structures in a given input instance significantly simplifies our problem since they have easy solutions. To this end, the following result by Deza [13] will serve as an important tool since it describes conditions under which a weak Δ -system actually becomes a strong one, that is, a sunflower (see also Jukna [23, Chapter 6, Theorem 6.2]).

Lemma 7 (Deza [13, Theorem 2]). *Let \mathcal{F} be an s -uniform weak Δ -system, that is, each set contains s elements. If $|\mathcal{F}| \geq s^2 - s + 2$, then \mathcal{F} is a sunflower.*

1	2	3	4	5	6	7	8	9	10
1	1		1					1	
1	1					1			
1	1	1							1
1	1				1		1		
1	1			1					
1	1								

Figure 4: Example of a matrix where the set system \mathcal{D} forms a sunflower with core $C = \{1, 2\}$ consisting of the first two columns. The six petals from top to bottom are $\{4, 9\}$, $\{7\}$, $\{3, 10\}$, $\{6, 8\}$, $\{5\}$ and \emptyset . Framed by thick lines is a set of columns that distinguish all rows.

The basic scheme for proving polynomial-time solvability of BINARY (α, β) -DISTINCT VECTORS for $\alpha \leq \beta < 2\lceil \alpha/2 \rceil + 2$ is the following: The bounds on the minimum and maximum pairwise row Hamming distances imply that the column systems \mathcal{W}_x for $x = \alpha, \dots, \beta$ form x -uniform weak Δ -systems. Using [Lemma 7](#), we then conclude that either the size of the instance is bounded by a constant depending on β only, or that the \mathcal{W}_x form sunflowers, which we can handle according to [Lemma 6](#).

As a final prerequisite, we prove the following easy but helpful lemma, concerning the intersection of sets with sunflowers.

Lemma 8. *Let $\lambda \in \mathbb{N}$, let \mathcal{F} be a sunflower with core C and let X be a set such that $|X \cap S| \geq \lambda$ for all $S \in \mathcal{F}$. If $|\mathcal{F}| > |X|$, then $\lambda \leq |C|$ and $|X \cap C| \geq \lambda$.*

Proof. Assume towards a contradiction that $|X \cap C| < \lambda$. Then X would intersect each of the $|\mathcal{F}| > |X|$ pairwise disjoint petals of \mathcal{F} , which is not possible. \square

We are now ready to prove the following theorem.

Theorem 9. BINARY (α, β) -DISTINCT VECTORS *is solvable*

- 1.) in $O(\min\{n, d\} \cdot nd)$ time if $\beta \leq \alpha + 1$, and
- 2.) in $O(n^3 d)$ time if α is odd and $\beta = \alpha + 2$.

We prove both statements of [Theorem 9](#) separately. As mentioned, the basic structures of both proofs are similar: We first partition the column system into uniform weak Δ -systems. Then, we consider each of the cases of which of the systems are sunflowers or of bounded size. Then, we leverage the preprocessing ([Lemma 5](#) and [Reduction Rule 1](#)) and our knowledge of solutions for sunflowers ([Lemma 6](#)) to show that only a small number of possible solutions are left. That is, the instances are essentially solved by the preprocessing routines and we can try out all remaining solutions to solve the instances in polynomial time. Showing that the remaining possible solutions are few seems more difficult for Statement (2.); thus, the proof of Statement (1.) can be seen as a “warm-up”.

Proof (**Theorem 9**, Statement (1.)). In the following, let $I := (S \in \{0,1\}^{n \times d}, k)$ be an instance of BINARY (α, β) -DISTINCT VECTORS for $\alpha \leq \beta \leq \alpha + 1$. Recall that we assume I to be already preprocessed according to **Lemma 5** and **Reduction Rule 1** in $O(\min\{n, d\} \cdot nd)$ time, that is, S contains the null row vector, say $s_n = \mathbf{0}$, no two column vectors are equal, which implies $d \leq 2^n$, and there are no inessential columns. We write $W_i := W_{s_i} = \{j \in [d] \mid s_{ij} = 1\}$ for the set of column indices j where row vector s_i equals 1, and we define $W_{ij} := W_i \cap W_j$. For $\omega \in [d]$, let $I_\omega := \{i \in [n] \mid w(s_i) = \omega\}$ denote the set of indices of the weight- ω rows and let $n_\omega := |I_\omega|$. For ease of presentation, we sometimes identify columns or rows and their corresponding indices.

For all $i \in [n - 1]$, we have

$$\Delta(s_i, s_n) = \Delta(s_i, \mathbf{0}) = w(s_i) \in \{\alpha, \alpha + 1\}.$$

Since also $\Delta(s_i, s_j) = w(s_i) + w(s_j) - 2|W_{ij}| \in \{\alpha, \alpha + 1\}$ for all $i \neq j \in [n - 1]$, the following properties can be derived:

$$\forall i, j \in I_\alpha, i \neq j : |W_{ij}| = \lfloor \alpha/2 \rfloor, \quad (1)$$

$$\forall i, j \in I_{\alpha+1}, i \neq j : |W_{ij}| = \lceil (\alpha + 1)/2 \rceil, \text{ and} \quad (2)$$

$$\forall i \in I_\alpha, j \in I_{\alpha+1} : |W_{ij}| = \lfloor (\alpha + 1)/2 \rfloor. \quad (3)$$

For example, let us prove Property (1). If $i, j \in I_\alpha$, then $2\alpha - 2|W_{ij}| \in \{\alpha, \alpha + 1\}$. If α is even, then, since $|W_{ij}|$ is an integer, $2\alpha - 2|W_{ij}| = \alpha$. Thus, $|W_{ij}| = \alpha/2 = \lfloor \alpha/2 \rfloor$. If α is odd, then $2\alpha - 2|W_{ij}| = \alpha + 1$ and, hence, $|W_{ij}| = (\alpha - 1)/2 = \lfloor \alpha/2 \rfloor$. This proves Property (1). The proofs for the remaining properties are analogous.

Property (1) implies that $\mathcal{W}_\alpha := \{W_i \mid i \in I_\alpha\}$ is an α -uniform weak Δ -system and Property (2) implies that $\mathcal{W}_{\alpha+1} := \{W_i \mid i \in I_{\alpha+1}\}$ is an $(\alpha + 1)$ -uniform weak Δ -system. Let $c := (\alpha + 1)^2 - (\alpha + 1) + 2$. We can assume that $\max\{n_\alpha, n_{\alpha+1}\} \geq c$ because otherwise $n \leq 2c$ is of constant size, and thus also $d \leq 2^n$ is of constant size (recall that we assume I to be preprocessed according to **Lemma 5**), which implies that I is constant-time solvable.

First, consider the case that $n_{\alpha+1} \geq c$. Then, by **Lemma 7**, it follows that $\mathcal{W}_{\alpha+1}$ is a sunflower with a core C of size $\lceil (\alpha + 1)/2 \rceil$ and petals $\widetilde{W}_i, i \in I_{\alpha+1}$, of size $\alpha + 1 - |C| \geq 1$. For each $i \in I_{\alpha+1}$ and each $j \in I_\alpha$, it follows by Property (3) and **Lemma 8** that $W_{ij} \subseteq C$, that is $\widetilde{W}_i \cap W_j = \emptyset$. Hence, for $x \in \widetilde{W}_i$, the column vector s_{*x} contains exactly one 1 (namely in the i -th row), that is, $s_{*x} = \mathbf{1}_{\{i\}}^n$. Thus, column x exactly distinguishes row i from all other rows. For $\alpha \geq 2$, each pair of rows differs in at least two columns. Thus, all rows in $S_{[d] \setminus \{x\}}$ are still distinct and column x is in fact inessential, which yields a contradiction. Hence, we can assume that $\alpha = 1$. Since, then, for all $i \in I_1$ and $j \in I_2$ we have $W_{ij} = W_i = C$, it follows that $n_1 = 1$. See **Figure 5a** for an illustrating example. The only possible solution is thus $K = \bigcup_{i \in [n-1]} W_i = [d]$.

If $n_{\alpha+1} < c$, then $n_\alpha \geq c$ holds and **Lemma 7** implies that \mathcal{W}_α is a sunflower with a core C of size $|C| = \lfloor \alpha/2 \rfloor$. If $(\alpha + 1)$ is even, then we have $\lfloor (\alpha + 1)/2 \rfloor > |C|$, and thus, by Property (3) and **Lemma 8**, it follows $n_{\alpha+1} = 0$ (see **Figure 5b**). Now, by **Lemma 6**, I is a yes-instance if and only if $k \geq n_\alpha$. If α is even, then $|C| = \lfloor (\alpha + 1)/2 \rfloor$,

1	1			
1		1		
1			1	
1				1
1				

(a)

1				
	1			
		1		
			1	
				1

(b)

Figure 5: Examples of two possible instances for the case $\alpha = 1, \beta = 2$.

and thus, Property (3) and Lemma 8 imply that $W_{ij} = C$ for all $i \in I_\alpha, j \in I_{\alpha+1}$. Note that $s_{*x} = \mathbf{1}_{[n-1]}^n$ for all $x \in C$, that is, column x exactly distinguishes row n from all others. Since α is even, hence $\alpha \geq 2$, it follows that column x is inessential, which again yields a contradiction. \square

Next, we show that BINARY (α, β) -DISTINCT VECTORS is solvable in $O(n^3 d)$ time if α is odd and $\beta = \alpha + 2$. We use the same notation as in the proof of Statement (1.).

Proof (Theorem 9, Statement (2.)). Since $\Delta(s_i, s_j) = w(s_i) + w(s_j) - 2|W_{ij}| \in \{\alpha, \alpha + 1, \alpha + 2\}$ holds for all $i \neq j \in [n]$, it follows that $\Delta(s_i, s_n) = w(s_i) \in \{\alpha, \alpha + 1, \alpha + 2\}$ holds for all $i \in [n - 1]$. By plugging in the respective values for $w(s_i)$ and $w(s_j)$ in the above formula for $\Delta(s_i, s_j)$, the following properties can be derived (in an analogous way as for Properties (1) to (3) in the proof of Statement (1.)):

$$\forall i, j \in I_\alpha, i \neq j : |W_{ij}| = \lfloor \alpha/2 \rfloor, \quad (1)$$

$$\forall i \in I_\alpha, j \in I_{\alpha+1} : |W_{ij}| \in \{\lfloor \alpha/2 \rfloor, \lceil \alpha/2 \rceil\}, \quad (2)$$

$$\forall i \in I_\alpha, j \in I_{\alpha+2} : |W_{ij}| = \lceil \alpha/2 \rceil, \quad (3)$$

$$\forall i, j \in I_{\alpha+1}, i \neq j : |W_{ij}| = \lceil \alpha/2 \rceil, \quad (4)$$

$$\forall i \in I_{\alpha+1}, j \in I_{\alpha+2} : |W_{ij}| \in \{\lceil \alpha/2 \rceil, \lceil \alpha/2 \rceil + 1\}, \quad (5)$$

$$\forall i, j \in I_{\alpha+2}, i \neq j : |W_{ij}| = \lceil \alpha/2 \rceil + 1. \quad (6)$$

Properties (1), (4), and (6) imply that \mathcal{W}_α , $\mathcal{W}_{\alpha+1}$, and $\mathcal{W}_{\alpha+2}$ are α -, $(\alpha + 1)$ -, and $(\alpha + 2)$ -uniform weak Δ -systems, respectively.

In the following, we denote by $U_\omega := \bigcup_{i \in I_\omega} W_i$ the index set of the columns where at least one weight- ω row vector equals 1. Let $c := (\alpha + 2)^2 - (\alpha + 2) + 2$. For each $x \in \{\alpha, \alpha + 1, \alpha + 2\}$, we either have $n_x < c$ or $n_x \geq c$. Overall, this gives eight possible cases, each of which we now show how to solve:

Case I ($n_\alpha < c, n_{\alpha+1} < c, n_{\alpha+2} < c$). In this case, the number of rows in S is upper-bounded by a constant depending on α , and thus, I is of overall constant size.

Case II ($n_\alpha \geq c, n_{\alpha+1} < c, n_{\alpha+2} < c$). Due to Lemma 7, family \mathcal{W}_α forms a sunflower. Let C with $|C| = \lfloor \alpha/2 \rfloor$ be the core of \mathcal{W}_α . For $\alpha = 1$, clearly, any solution K contains all column indices from U_1 in order to distinguish the weight-1 rows from the null vector. Since $|U_2 \cup U_3| \leq 2n_2 + 3n_3$ is upper-bounded by a constant, the number

of possible subsets $K' \subseteq U_2 \cup U_3$ is also upper-bounded by a constant. Thus, we only have to check a constant number of choices $K = U_1 \cup K'$. For $\alpha \geq 3$, the size of a petal \widetilde{W}_i , $i \in I_\alpha$, is $|\widetilde{W}_i| = |W_i| - |C| = \alpha - \lfloor \alpha/2 \rfloor = \lceil \alpha/2 \rceil \geq 2$. Since the petals are pairwise disjoint, it follows that, for each petal \widetilde{W}_i , there exists a $j \in I_{\alpha+1} \cup I_{\alpha+2}$ such that $\widetilde{W}_i \cap W_j \neq \emptyset$. Otherwise, the column vectors corresponding to the indices in a petal \widetilde{W}_i are all equal to $\mathbf{1}_{\{i\}}^n$, that is, at least one of them is inessential, which is a contradiction. Since $|U_{\alpha+1} \cup U_{\alpha+2}|$ is upper-bounded by a constant depending on α , also the number n_α of petals in \mathcal{W}_α is upper-bounded by a constant, which yields an overall constant size of I .

Case III ($n_\alpha < c$, $n_{\alpha+1} \geq c$, $n_{\alpha+2} < c$). By Property (4) and Lemma 7, family $\mathcal{W}_{\alpha+1}$ forms a sunflower with a core C of size $|C| = \lceil \alpha/2 \rceil$. The size of each petal \widetilde{W}_i , $i \in I_{\alpha+1}$, is thus $|\widetilde{W}_i| = \lceil \alpha/2 \rceil$. Hence, for $\alpha \geq 3$, the same arguments as in Case (II) hold. For $\alpha = 1$, any solution K can be written as $K = C' \cup U_1 \cup K_2 \cup K_3$, where $C' \subseteq C$, $K_2 \subseteq U_2 \setminus C$ and $K_3 \subseteq U_3$. Note that $|C|$ and $|U_3|$ are upper-bounded by a constant. Hence, the number of different subsets C' and K_3 is also a constant. Since $|\widetilde{W}_i| = 1$ holds for all $i \in I_2$, we have $|U_2 \setminus C| = n_2$. From Lemma 6, it follows that $|K_2| \geq n_2 - 1$. The overall number of possible choices for K_2 , and thus for K , is in $O(n)$.

Case IV ($n_\alpha < c$, $n_{\alpha+1} < c$, $n_{\alpha+2} \geq c$). By Lemma 7, family $\mathcal{W}_{\alpha+2}$ forms a sunflower with core C of size $|C| = \lceil \alpha/2 \rceil + 1$. The size of each petal \widetilde{W}_i , $i \in I_{\alpha+2}$, is thus $|\widetilde{W}_i| = \lceil \alpha/2 \rceil$. Hence, for $\alpha \geq 3$, the same arguments as in Case (II) hold. For $\alpha = 1$, any solution K can be written as $K = C' \cup U_1 \cup K_2 \cup K_3$, where $C' \subseteq C$, $K_2 \subseteq U_2$ and $K_3 \subseteq U_3 \setminus C$. Note that $|C|$ and $|U_2|$ are upper-bounded by a constant. Hence, the number of different subsets C' and K_2 is also a constant. Lemma 6 implies that $|K_3| \geq n_3 - 1$. Since $|U_3 \setminus C| = n_3$, this yields an overall number of $O(n)$ possible choices for K .

Case V ($n_\alpha \geq c$, $n_{\alpha+1} \geq c$, $n_{\alpha+2} < c$). Due to Lemma 7, family \mathcal{W}_α forms a sunflower with a core C of size $|C| = \lfloor \alpha/2 \rfloor$ and $\mathcal{W}_{\alpha+1}$ forms a sunflower with core C' of size $|C'| = \lceil \alpha/2 \rceil$. First, note that Property (3) implies $|W_{ij}| = \lceil \alpha/2 \rceil > |C|$ for all $i \in I_\alpha$, $j \in I_{\alpha+2}$, which is not possible due to Lemma 8. Thus, it follows $n_{\alpha+2} = 0$. Moreover, since Property (2) implies $|W_{ij}| \geq |C|$ for all $i \in I_\alpha$ and $j \in I_{\alpha+1}$, Lemma 8 yields $C \subseteq W_{ij}$, and thus $C \subset C'$. Hence, all column vectors in C equal $\mathbf{1}_{[n-1]}^n$, which yields a contradiction for $\alpha \geq 3$ because the columns in C are then inessential. For $\alpha = 1$, any solution K can be written as $K = C'' \cup U_1 \cup K_2$, where $C'' \subseteq C'$ and $K_2 \subseteq U_2$. By Lemma 6, we know that $|K_2| \geq n_2 - 1$. Since $|C'| = 1$ and $|U_2 \setminus C'| = n_2$, there are $O(n)$ possible choices for K .

Case VI ($n_\alpha \geq c$, $n_{\alpha+1} < c$, $n_{\alpha+2} \geq c$). This case is not possible since we showed in Case V that $n_\alpha \geq c$ implies $n_{\alpha+2} = 0$.

Case VII ($n_\alpha \geq c$, $n_{\alpha+1} \geq c$, $n_{\alpha+2} \geq c$). This case is also not possible, see Case V.

Case VIII ($n_\alpha < c$, $n_{\alpha+1} \geq c$, $n_{\alpha+2} \geq c$). From Lemma 7 and Properties (4) and (6), respectively, it follows that $\mathcal{W}_{\alpha+1}$ forms a sunflower with a core C of size $|C| = \lceil \alpha/2 \rceil$ and $\mathcal{W}_{\alpha+2}$ forms a sunflower with core C' of size $|C'| = \lceil \alpha/2 \rceil + 1$. Moreover, as in Case V, Property (5) and Lemma 8 imply $C \subset C'$.

If $\alpha = 1$, then any solution can be written as $K = U_1 \cup C'' \cup K_2 \cup K_3$, where $C'' \subseteq C'$, $K_2 \subseteq U_2 \setminus C$ and $K_3 \subseteq U_3 \setminus C'$. Since $|C'| = \lceil \alpha/2 \rceil + 1$, $|U_2 \setminus C| = n_2$, $|U_3 \setminus C'| = n_3$, and,

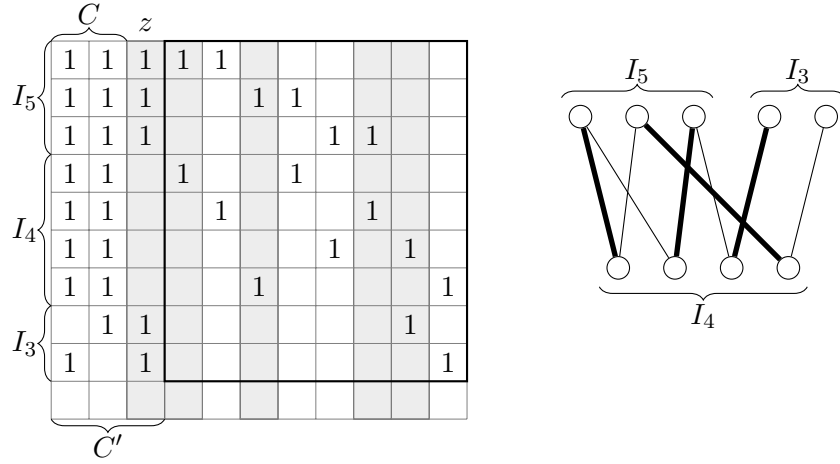


Figure 6: An instance for the case $\alpha = 3$, $\beta = 5$ (left). The submatrix framed by the thick rectangle defines a bipartite graph (right). An optimal solution is highlighted in gray. Note that the columns in the solution correspond to a matching in the bipartite graph that saturates I_4 , represented by the thick lines.

by Lemma 6, $|K_2| \geq n_2 - 1$ and $|K_3| \geq n_3 - 1$, it follows that there are $O(n^2)$ possible choices for K .

For $\alpha \geq 3$, we show that the matrix S —recall that it is reduced with respect to Reduction Rule 1—has a specific structure, depicted in Figure 6. Namely, we claim that

(a) if $W_{ij} \setminus C' \neq \emptyset$ for $i \neq j$, then $i \in I_{\alpha+1}$ and $j \in I_\alpha \cup I_{\alpha+2}$, and

(b) the unique column vector s_{*z} with $z \in C \setminus C'$ equals $\mathbf{1}_{I_\alpha \cup I_{\alpha+2}}^n$.

Claim (a) implies that each column in $[d] \setminus C'$ contains at most two 1's (naturally, any column contains at least one 1). We will see that all columns in $[d] \setminus C'$ contain exactly two 1's and, hence, that they define the edges of a bipartite graph with the two partite vertex sets $I_{\alpha+1}$ and $I_\alpha \cup I_{\alpha+2}$. We find a matching that saturates $I_{\alpha+1}$ in this bipartite graph and show that the columns corresponding to the matching edges along with column z are an optimal solution.

To show Claim (a), observe that, if $i \neq j \in I_{\alpha+2}$, then $W_{ij} \setminus C' = \emptyset$ as $\mathcal{W}_{\alpha+2}$ is a sunflower with core C' . Likewise, if $i \neq j \in I_{\alpha+1}$, then $W_{ij} \setminus C' = \emptyset$ because $\mathcal{W}_{\alpha+1}$ is a sunflower with core C and $C \subset C'$. It hence suffices to show that $W_{ij} \setminus C' = \emptyset$ in the case that either both $i, j \in I_\alpha$ or $i \in I_\alpha$ and $j \in I_{\alpha+2}$. To see the latter, note that Property (3) and Lemma 8 imply $|W_i \cap C'| = \lceil \alpha/2 \rceil = |C'| - 1$, for all $i \in I_\alpha$, that is, we even have $W_{ij} \subset C'$ for all $i \in I_\alpha$, $j \in I_{\alpha+2}$. Now, it only remains to show $W_{ij} \subseteq C'$ for $i, j \in I_\alpha$, $i \neq j$. We derived above that $|W_i \cap C'| = |W_j \cap C'| = \lceil \alpha/2 \rceil = |C'| - 1$. Thus, $|(W_i \cap C') \cap (W_j \cap C')| = |W_{ij} \cap C'| \geq |C'| - 2 = \lfloor \alpha/2 \rfloor$. By Property (1), $|W_{ij}| \leq \lfloor \alpha/2 \rfloor$, which implies $W_{ij} \subseteq C'$. Hence, $W_{ij} \setminus C' = \emptyset$, completing the proof of Claim (a).

Let us next prove Claim (b), that is, $s_{*z} = \mathbf{1}_{I_\alpha \cup I_{\alpha+2}}^n$ where z is the unique column in $C \setminus C'$. Assume the contrary, that is, either a row in I_α has a 0 at entry z or a

row in $I_{\alpha+1}$ has a 1 at entry z . Let us first show that $s_{iz} = 0$, that is, $z \notin W_i$ is impossible for a row $i \in I_\alpha$. Using $|W_i \cap C'| = |C'| - 1$, it follows that $C \subset W_i$. Then, for all $j \in I_{\alpha+1} \cup I_{\alpha+2}$, it holds $W_{ij} \setminus C = \emptyset$ since otherwise either Property (2) or Property (3) is violated. Let us show that $W_{ij} \setminus C = \emptyset$ also for all $j \neq i \in I_\alpha$. Recall that $W_{ij} \setminus C' = \emptyset$, as shown above. By assumption $z \notin W_{ij}$, yielding $W_{ij} \setminus C = W_{ij} \setminus C' = \emptyset$. But then, the columns in $W_i \setminus C$ equal $\mathbf{1}_{\{i\}}^n$. Note that $|W_i \setminus C| \geq \alpha - \lfloor \alpha/2 \rfloor \geq 2$ (recall that $\alpha \geq 3$). Hence, there is at least one inessential column, a contradiction. By the same arguments (using Properties (4) and (5)), we can infer that there is no $i \in I_{\alpha+1}$ such that $z \in W_i$, that is, $C' \subset W_i$. Hence, for $z \in C' \setminus C$, it holds $s_{*z} = \mathbf{1}_{I_\alpha \cup I_{\alpha+2}}^n$, proving Claim (b). Note that column z distinguishes all rows in $I_{\alpha+1}$ from all rows in $I_\alpha \cup I_{\alpha+2}$.

To finish the proof of Case VIII, we need one more observation about the rows in I_α , namely that $n_\alpha = \lceil \alpha/2 \rceil$. Assume the contrary, that is, since we have $|W_i \cap C'| = |C'| - 1 = |C| = \lceil \alpha/2 \rceil$ and $z \in W_i$ for all $i \in I_\alpha$, there exists an $x \in C$ such that $x \in W_i$ for all $i \in I_\alpha$. Then, $s_{*x} = \mathbf{1}_{[n-1]}^n$ and thus, column x is inessential, which is not possible. Hence, for each $x \in C$, there exists an $i \in I_\alpha$ such that $x \notin W_i$. Since $x \in C'$ and $|W_i \cap C'| = |C'| - 1$ it follows that $W_i \cap C' = C' \setminus \{x\}$. Therefore, we have $n_\alpha = |C| = \lceil \alpha/2 \rceil$.

We now derive a solution from the abovementioned bipartite graph. Consider the columns in $[d] \setminus C'$. Clearly, if one of these columns contains only one 1, then this column is inessential, which yields a contradiction. Thus, each column contains at least two 1's. Using Claim (a), each of the columns also has at most two 1's. Also, after preprocessing, no two columns are equal. Thus, the submatrix $S[[n-1], [d] \setminus C']$ (framed by thick lines in Figure 6) is the incidence matrix of a bipartite graph G , where the rows correspond to the vertices (partitioned into $I_{\alpha+1}$ and $I_\alpha \cup I_{\alpha+2}$) and the columns define the edges. Moreover, each vertex $i \in I_{\alpha+2}$ has degree $|W_i \setminus C'| = \lceil \alpha/2 \rceil$, since $s_{*z} = \mathbf{1}_{I_\alpha \cup I_{\alpha+2}}^n$ also each vertex $i \in I_{\alpha+1}$ has degree $\lceil \alpha/2 \rceil$, and, since each row $i \in I_\alpha$ has $|W_i \cap C'| = \lceil \alpha/2 \rceil$ (as derived above), each vertex $i \in I_\alpha$ has degree $|W_i \setminus C'| = \lfloor \alpha/2 \rfloor$ in G . We can now use Hall's theorem [2], to show that there exists a matching in G that saturates $I_{\alpha+1}$, that is, a subset $M \subseteq [d] \setminus C'$ of $n_{\alpha+1}$ columns such that $|W_i \cap M| = 1$ for all $i \in I_{\alpha+1}$ and $|W_i \cap M| \leq 1$ for all $i \in I_\alpha \cup I_{\alpha+2}$.³ Indeed, taking any subset $T \subseteq I_{\alpha+1}$ of vertices, consider the set $N_G(T) \subseteq I_\alpha \cup I_{\alpha+2}$ of neighbors of T . Since the vertices in $N_G(T)$ have at most the degree of any vertex in T , we have $|N_G(T)| \geq |T|$. Hence, the precondition of Hall's theorem is satisfied. Thus, M exists as claimed.

We now claim that $K := M \cup \{z\}$ with $|K| = n_{\alpha+1} + 1$ is an optimal solution (highlighted in gray in Figure 6). First, regarding K being a solution, since G is bipartite, we have $n_{\alpha+2} \lceil \alpha/2 \rceil + n_\alpha \lfloor \alpha/2 \rfloor = n_{\alpha+1} \lceil \alpha/2 \rceil$. From this, we can infer $n_{\alpha+1} = n_{\alpha+2} + \lfloor \alpha/2 \rfloor = |I_\alpha \cup I_{\alpha+2}| - 1$. Thus, as M saturates $I_{\alpha+1}$, there exists exactly one $j \in I_\alpha \cup I_{\alpha+2}$ such that $W_j \cap M = \emptyset$. Using this, it is not hard to check that K is a solution.

Regarding optimality, it remains to show that there is no solution of size $n_{\alpha+1}$. This can be seen as follows: Lemma 6 implies that any solution K intersects at least $n_{\alpha+1} - 1$

³Hall's theorem asserts that, for a bipartite graph $G = (X \cup Y, E)$, there exists an X -saturating matching if and only if $|T| \leq |N_G(T)|$ holds for each subset $T \subseteq X$.

of the petals of $\mathcal{W}_{\alpha+1}$. If K intersects each petal, then, as $n_{\alpha+1} = n_{\alpha} + n_{\alpha+2} - 1$, there exists a $j \in I_{\alpha} \cup I_{\alpha+2}$ such that $K \cap W_j = \emptyset$, which is not possible. Otherwise, if K intersects exactly all but one of the petals, then there exists an $i \in I_{\alpha+1}$ and also $j \neq j' \in I_{\alpha} \cup I_{\alpha+2}$ such that $K \cap W_i \setminus C' = \emptyset$, $K \cap W_j \setminus C' = \emptyset$ and $K \cap W_{j'} \setminus C' = \emptyset$. In order to distinguish row i from the null vector, K has to contain a column from C . But it is not possible to pairwise distinguish all three rows i , j , and j' from each other with just one column. Hence, K is indeed optimal and I is a yes-instance if and only if $k \geq n_{\alpha+1} + 1$.

As regards the running time, observe that the maximum number of candidate solutions we have to test in any of the above cases is in $O(n^2)$. Checking whether a subset of column indices is a solution can be done in $O(nd)$ time via lexicographical sorting of the rows. This yields an overall running time in $O(n^3d)$ which also subsumes the $O(\min\{n, d\} \cdot nd)$ time for the preprocessing. \square

4 Distinct Vectors on General Matrices

In the last section, we have seen, among other results, that DISTINCT VECTORS is NP-complete and W[1]-hard with respect to the number t of columns to be deleted even if the input alphabet is binary and the pairwise Hamming distance of the row vectors is bounded by four (Corollary 2). Note, however, that the parameterized complexity with respect to the number k of retained columns for binary alphabets remained open. In this section, we first show that HITTING SET parameterized by the solution size (which is W[2]-complete [15]) is parameterized reducible to DISTINCT VECTORS for alphabets of unbounded size, showing that DISTINCT VECTORS is W[2]-hard with respect to k (Theorem 10). Nevertheless, we show later in this section some tractability results even for larger alphabets. For example, we give a problem kernel with respect to the combined parameter alphabet size $|\Sigma|$ and number k of retained columns (Theorem 12). Note that this result implies that DISTINCT VECTORS is fixed-parameter tractable with respect to k for any alphabet of constant size.

Theorem 10. *DISTINCT VECTORS is W[2]-hard with respect to the number k of retained columns.*

Proof. We give a parameterized reduction from the W[2]-complete HITTING SET problem parameterized by solution size k .

HITTING SET

Input: A finite universe U , a collection \mathcal{C} of subsets of U , and a nonnegative integer k .

Question: Is there a subset $K \subseteq U$ with $|K| \leq k$ such that K contains at least one element from each subset in \mathcal{C} ?

Given an instance (U, \mathcal{C}, k) of HITTING SET with $U = \{u_1, \dots, u_m\}$ and $\mathcal{C} = \{C_1, \dots, C_n\}$, we define the DISTINCT VECTORS instance (S, k') where $k' := k$ and the $(n+1) \times m$

$U = \{1, \dots, 6\}$	
$C_1 = \{1, 2, 3\}$	C_1
$C_2 = \{3, 4\}$	C_2
$C_3 = \{1, 3, 6\}$	C_3
$C_4 = \{1, 2, 4, 5\}$	C_4
$C_5 = \{1, 5, 6\}$	C_5

	1	2	3	4	5	6
C_1	1	1	1			
C_2			2	2		
C_3	3		3			3
C_4	4	4		4	4	
C_5	5				5	5

Figure 7: Example of a HITTING SET instance (left) and the constructed matrix (right). The hitting set $K = \{3, 5\}$ is indicated by thick lines.

matrix S is defined as

$$s_{ij} := \begin{cases} i, & u_j \in C_i \\ 0, & u_j \notin C_i \end{cases}$$

for all $i \in [n]$, $j \in [m]$ and $s_{n+1} := \mathbf{0}$. This instance is polynomial-time computable. An example is depicted in [Figure 7](#). If $K \subseteq U$ is a solution of (U, \mathcal{C}, k) , then $K \cap C_i \neq \emptyset$ holds for all $C_i \in \mathcal{C}$, and thus, for each row s_i , there exists a column j corresponding to some element $u_j \in K$ such that $s_{ij} = i$. Since no other row contains an entry equal to i , it follows that row s_i is distinct from all other rows in S . Conversely, in order to distinguish row s_i from $s_{(n+1)} = \mathbf{0}$, any solution K' of (S, k') has to contain a column index j such that $s_{ij} \neq 0$. This implies that the subset $\{u_j \mid j \in K'\} \subseteq U$ contains at least one element of each C_i and is thus a solution of the original instance. Finally, note that this is a parameterized reduction since $k' = k$. \square

Chen et al. [8] showed that HITTING SET cannot be solved in $|U|^{o(k)} \cdot |I|^{O(1)}$ time, unless $\text{FPT} = \text{W}[1]$. Since the reduction from HITTING SET yields an instance with $d = |U|$ columns and solution size k in polynomial time, the following corollary is immediate.

Corollary 11. *If $\text{FPT} \neq \text{W}[1]$, then DISTINCT VECTORS cannot be solved in $d^{o(k)} \cdot |I|^{O(1)}$ time.*

On the positive side, DISTINCT VECTORS can trivially be solved by trying all subsets of column indices of size k within $d^k \cdot |I|^{O(1)}$ time.

Although [Theorem 10](#) shows that DISTINCT VECTORS is $\text{W}[2]$ -hard with respect to the parameter k , we can provide a problem kernel for DISTINCT VECTORS if we additionally consider the input alphabet size $|\Sigma|$ as a second parameter. The size of the problem kernel is superexponential in the combined parameter $(|\Sigma|, k)$. Clearly, a problem kernel of polynomial size would be desirable. However, polynomial-size problem kernels do not exist even with the additional parameter number n of rows, unless $\text{NP} \subseteq \text{coNP}/\text{poly}$, which would imply a collapse of the polynomial hierarchy in complexity theory, which is widely believed not to be the case.

Theorem 12. *For DISTINCT VECTORS,*

- 1.) *there exists an $O(|\Sigma|^{|\Sigma|^k + k} / |\Sigma|! \cdot \log |\Sigma|)$ -size problem kernel computable in $O(d^2 n^2)$ time and*

2.) unless $\text{NP} \subseteq \text{coNP}/\text{poly}$, there is no polynomial-size problem kernel with respect to the combined parameter $(n, |\Sigma|, k)$.

Proof. 1.) Since S contains n rows, it follows that at least $k \geq \lceil \log_{|\Sigma|} n \rceil$ columns are required to distinguish all rows, otherwise we simply return a trivial no-instance. Thus, we have $n \leq |\Sigma|^k$. Moreover, note that each column partitions the rows into at most $|\Sigma|$ non-empty subsets (all rows with identical values form a subset of the partition). We use the following simple data reduction rule: If column j partitions the rows *finer* than column j' (that is, each set in the partition of j is a subset of a set in the partition of j'), then delete column j' . This rule clearly is correct since column j distinguishes all pairs of rows that are distinguishable by column j' . Exhaustive application of the above rule requires $O(d^2 n^2)$ arithmetic operations (checking for all $j, j' \in [d]$ whether $s_{ij'} \neq s_{i'j'} \Rightarrow s_{ij} \neq s_{i'j}$ holds for all $i, i' \in [n]$). It follows that for each remaining pair of columns, the partition of one is not finer than the partition of the other. Thus, we can bound the number d of columns from above by $|\Sigma|^n / |\Sigma|!$. (More precisely, d is bounded by the cardinality of a maximum *antichain*, that is, a set of partitions being pairwise incomparable with respect to the “finer than” order, in the *partition lattice* of an n -element set up to the $|\Sigma|$ -th level, see Grätzer [21, Chapter IV.4] for details). The overall size of S is thus in

$$O(nd \cdot \log |\Sigma|) = O(|\Sigma|^{|\Sigma|^k + k} / |\Sigma|! \cdot \log |\Sigma|),$$

which yields a problem kernel with respect to the combined parameter $(|\Sigma|, k)$.

2.) We give a lower bound on the size of a problem kernel based on a result by Dom et al. [14], who showed that there is no polynomial-size problem kernel for SET COVER with respect to the combined parameter $(|U|, k)$, unless $\text{NP} \subseteq \text{coNP}/\text{poly}$ (which implies the collapse of the polynomial hierarchy).

SET COVER

Input: A finite universe U , a collection \mathcal{C} of subsets of U , and a nonnegative integer k .

Question: Is there a subset $S \subseteq \mathcal{C}$ with $|S| \leq k$ such that each element of U is contained in at least one subset in S ?

The reduction from HITTING SET in the proof of Theorem 10 can be used to obtain a reduction from SET COVER, when first transforming SET COVER into HITTING SET in the common way [1], that is, the universe of the HITTING SET instance is \mathcal{C} and for each element $u \in U$, there is the subset $\{C \in \mathcal{C} \mid u \in C\}$. The resulting DISTINCT VECTORS instance consists of a matrix with $n = |U|$ rows over an alphabet of size $|\Sigma| = |U| + 1$ and a sought solution size k . Since SET COVER is NP-complete, there is a polynomial-time many-one reduction from DISTINCT VECTORS to SET COVER and, hence, a polynomial-size kernel for DISTINCT VECTORS would imply a polynomial-size kernel for SET COVER: simply transform the SET COVER instance into a DISTINCT VECTORS instance, kernelize, and transform back. \square

Observe the gap between the superpolynomial lower bound and the superexponential upper bound on the problem kernel size in Theorem 12, which leaves a significant gap.

Proving the (non)-existence of a $|\Sigma|^{O(k)}$ -size problem kernel, for example, would be an interesting result.

We now move on to parameterizing by the maximum pairwise row Hamming distance H . Recall [Definition 2](#), where, for a matrix $S \in \Sigma^{n \times d}$, we defined $H := \max_{i \neq j \in [n]} \Delta(s_i, s_j)$. In this case, every pair of rows in S differs in at most H columns, which yields a kernelization and also a fairly simple approximation algorithm by a reduction from DISTINCT VECTORS to HITTING SET:

Theorem 13. *Let H be the maximum pairwise row Hamming distance of the input matrix. Then, DISTINCT VECTORS*

- 1.) *is linear-time factor- H approximable and*
- 2.) *admits an $O(g(H, k)^2 \log g(H, k))$ -size problem kernel which can be computed in $O(d^2 + n^2 \max\{d \log d, dn^2\})$ time, where $g(H, k) := H! \cdot H^{H+1} \cdot (k+1)^H$.*

Proof. The idea for both results is to define a polynomial-time parameterized many-one reduction from DISTINCT VECTORS to H -HITTING SET, which is the special case of HITTING SET where each input set has cardinality at most H . We can then apply known kernelization and approximation algorithms to the H -HITTING SET instance. The reduction works as follows: Given an instance (S, k) of DISTINCT VECTORS, the H -HITTING SET instance (U, \mathcal{C}, k') is defined as

$$U := [d], \mathcal{C} := \{C_{ij} \subseteq U \mid i \neq j \in [n]\}, \text{ where } C_{ij} := \{u \in U \mid s_{iu} \neq s_{ju}\},$$

and $k' := k$. Note that $|C_{ij}| \leq H$ holds for all $i \neq j$. This reduction requires $O(n^2 d)$ arithmetic operations. It is correct since $K \subseteq [d]$ with $|K| \leq k$ is a solution of (S, k) if and only if for every pair of rows in S there is at least one column in K in which both rows have different values. This is equivalent to the situation that K contains at least one element from each C_{ij} in \mathcal{C} , which implies that K is a solution of (U, \mathcal{C}, k) .

We now prove the two statements of the theorem using the above reduction.

1.) A factor- H approximation algorithm repeatedly adds a so far unhit subset to the hitting set.

2.) Let (U, \mathcal{C}, k) with $|U| = d$ and $|\mathcal{C}| \in O(n^2)$ be the H -HITTING SET instance resulting from the above reduction. We apply a H -HITTING SET kernelization due to van Bevern [3] in order to obtain in $O(Hd + H \log H \cdot n^2 + Hn^4)$ time an instance (U', \mathcal{C}', k) , where $|U'|$ and $|\mathcal{C}'|$ are at most $g(H, k)$. In order to obtain a problem kernel for DISTINCT VECTORS, we transform the instance (U', \mathcal{C}', k) back by the reduction from the proof of [Theorem 10](#). We end up with a DISTINCT VECTORS instance (S', k') with $k' = k$ in $O(|U'| \cdot |\mathcal{C}'|) = O(n^2 d)$ time. Since (U', \mathcal{C}', k) is an instance of H -HITTING SET, it follows that each row in S' differs from $\mathbf{0}$ in at most H columns. Thus, each pair of rows in S' differs in at most $H' \leq 2H$ columns. Note that k' and H' depend only on k and h , which also holds for the overall size of S' , which is in $O(|U'| \cdot |\mathcal{C}'| \log |\mathcal{C}'|) = O(g(H, k)^2 \log(g(H, k)))$. Moreover, the overall running time is in $O(n^2 d + Hd + H \log H \cdot n^2 + Hn^4)$, which gives a problem kernel. \square

In this section, we have seen that DISTINCT VECTORS can basically be regarded as a special HITTING SET problem. HITTING SET in general is $W[2]$ -complete [15] with respect to the solution size, but DISTINCT VECTORS is fixed-parameter tractable with respect to the solution size for constant-size alphabets (Theorem 12). Thus, the set systems induced by constant-size alphabet instances of DISTINCT VECTORS involve a certain structure (that is, the number of subsets is exponentially upper-bounded in the size of the solution) that makes them somewhat easier to solve. Generalizing the analysis of the structure as we did in Section 3 for binary alphabets to arbitrary alphabets deserves further investigation.

5 Conclusion

We conclude with a few challenges for future research. Based on pairwise minimum and maximum Hamming distances, we proved a complexity dichotomy for DISTINCT VECTORS restricted to binary matrices. We leave generalizations of the polynomial-time solvable cases to non-binary alphabets as a major open question. A further interesting question is whether one can close the gap between the doubly-exponential upper and the superpolynomial lower bound for the size of the problem kernel for DISTINCT VECTORS parameterized by the combined parameter “alphabet size and number of remaining columns”.

From a combinatorial point of view, the study of “vector problems” in general seems to be a fertile but little researched area for parameterized complexity studies. Another example of a parameterized complexity analysis for a vector problem deals with the explanation of integer vectors by few homogenous segments [6]. Finally, on a more general scale, it seems that parameterized complexity analysis is a promising tool to better assessing the computational complexity of some machine learning problems such as combinatorial feature selection; our work is among the first contributions in this so far widely neglected research direction.

References

- [1] G. Ausiello, A. D’Atri, and M. Protasi. Structure preserving reductions among convex optimization problems. *Journal of Computer and System Sciences*, 21(1): 136–153, 1980. 21
- [2] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer, 2009. 18
- [3] R. van Bevern. Towards optimal and expressive kernelization for d -hitting set. *Algorithmica*, 70(1):129–147, 2014. 22
- [4] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997. 2

- [5] A. Brandstädt and R. Mosca. On distance-3 matchings and induced matchings. *Discrete Applied Mathematics*, 159(7):509–520, 2011. [6](#)
- [6] R. Brederick, J. Chen, S. Hartung, C. Komusiewicz, R. Niedermeier, and O. Suchý. On explaining integer vectors by few homogenous segments. *Journal of Computer and System Sciences*, 81(4):766–782, 2015. [23](#)
- [7] M. Charikar, V. Guruswami, R. Kumar, S. Rajagopalan, and A. Sahai. Combinatorial feature selection problems. In *Proceedings of the 41st IEEE Annual Symposium on Foundations of Computer Science (FOCS 2000)*, pages 631–640, 2000. [1](#), [2](#)
- [8] J. Chen, B. Chor, M. Fellows, X. Huang, D. Juedes, I. A. Kanj, and G. Xia. Tight lower bounds for certain parameterized NP-hard problems. *Information and Computation*, 201(2):216–231, 2005. [20](#)
- [9] C. Cotta and P. Moscato. The k -feature set problem is W[2]-complete. *Journal of Computer and System Sciences*, 67(4):686–690, 2003. [3](#)
- [10] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015. [4](#)
- [11] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, and M. W. Mahoney. Feature selection methods for text classification. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007)*, pages 230–239, 2007. [2](#)
- [12] S. Davies and S. Russell. NP-completeness of searches for smallest possible feature sets. In *AAAI Symposium on Intelligent Relevance*, pages 37–39, 1994. [3](#)
- [13] M. Deza. Solution d’un problème de Erdős-Lovász. *Journal of Combinatorial Theory, Series B*, 16(2):166–167, 1974. [12](#)
- [14] M. Dom, D. Lokshtanov, and S. Saurabh. Kernelization lower bounds through colors and IDs. *ACM Transactions on Algorithms*, 11(2):13:1–13:20, 2014. [21](#)
- [15] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013. [3](#), [4](#), [19](#), [23](#)
- [16] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006. [4](#)
- [17] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003. [2](#)
- [18] V. Froese. Combinatorial feature selection: Parameterized algorithms and complexity. Master’s thesis, TU Berlin, 2012. [1](#)
- [19] V. Froese, R. van Bevern, R. Niedermeier, and M. Sorge. A parameterized complexity analysis of combinatorial feature selection problems. In *Proceedings of the 38th International Symposium on Mathematical Foundations of Computer Science*

- (*MFCS 2013*), volume 8087 of *Lecture Notes in Computer Science*, pages 445–456. Springer, 2013. [1](#)
- [20] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979. [4](#)
 - [21] G. Grätzer. *General Lattice Theory*. Birkhäuser, 2003. [21](#)
 - [22] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003. [2](#)
 - [23] S. Jukna. *Extremal Combinatorics*. Springer, 2011. [11](#), [12](#)
 - [24] D. Koller and M. Sahami. Towards optimal feature selection. In *Proceedings of the 13th International Conference on Machine Learning (ICML 1996)*, pages 284–292, 1996. [1](#)
 - [25] H. Moser and D. M. Thilikos. Parameterized complexity of finding regular induced subgraphs. *Journal of Discrete Algorithms*, 7(2):181–190, 2009. [7](#)
 - [26] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. [4](#)
 - [27] C. H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994. [4](#)
 - [28] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic, 1991. [2](#)
 - [29] A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. In R. Slowinski, editor, *Intelligent Decision Support—Handbook of Applications and Advances of the Rough Sets Theory*, pages 331–362. Kluwer Academic, 1992. [2](#)